



Article Large-Area Full-Coverage Remote Sensing Image Collection Filtering Algorithm for Individual Demands

Boce Chu ^{1,2}, Feng Gao ^{2,3,*}, Yingte Chai ², Yu Liu ^{2,3}, Chen Yao ², Jinyong Chen ², Shicheng Wang ², Feng Li ² and Chao Zhang ²

- School of Electronics and Information Engineering, Beihang University, Beijing 100191, China; bocc012628077@163.com
- ² Key Laboratory of Aerospace Information Applications of CETC, Shijiazhuang 050081, China; wmeirui@163.com (Y.C.); csu_fdsun@163.com (Y.L.); SenoLee@whu.edu.cn (C.Y.); 18222961025@163.com (J.C.); hui_tian@163.com (S.W.); hcx624934141@163.com (F.L.); wangtao2114@mails.jlu.edu.cn (C.Z.)

Abstract: Remote sensing is the main technical means for urban researchers and planners to effec-

- ³ School of Electronic Engineering, Xidian University, Xi'an 710071, China
- * Correspondence: 15511267177@163.com

tively observe targeted urban areas. Generally, it is difficult for only one image to cover a whole urban area and one image cannot support the demands of urban planning tasks for spatial statistical analysis of a whole city. Therefore, people often artificially find multiple images with complementary regions in an urban area on the premise of meeting the basic requirements for resolution, cloudiness, and timeliness. However, with the rapid increase of remote sensing satellites and data in recent years, time-consuming and low performance manual filter results have become more and more unacceptable. Therefore, the issue of efficiently and automatically selecting an optimal image collection from massive image data to meet individual demands of whole urban observation has become an urgent problem. To solve this problem, this paper proposes a large-area full-coverage remote sensing image collection filtering algorithm for individual demands (LFCF-ID). This algorithm achieves a new image filtering mode and solves the difficult problem of selecting a full-coverage remote sensing image collection from a vast amount of data. Additionally, this is the first study to achieve fullcoverage image filtering that considers user preferences concerning spatial resolution, timeliness, and cloud percentage. The algorithm first quantitatively models demand indicators, such as cloudiness, timeliness, resolution, and coverage, and then coarsely filters the image collection according to the ranking of model scores to meet the different needs of different users for images. Then, relying on map gridding, the image collection is genetically optimized for individuals using a genetic algorithm (GA), which can quickly remove redundant images from the image collection to produce the final filtering result according to the fitness score. The proposed method is compared with manual filtering and greedy retrieval to verify its computing speed and filtering effect. The experiments show that the proposed method has great speed advantages over traditional methods and exceeds the results of manual filtering in terms of filtering effect.

Keywords: remote sensing image; filtering algorithm; individual demand; urban planning; genetic algorithm

1. Introduction

To meet the needs of urban planning, land surveying, and other applications for large-scale regional observation, remote sensing with high spatial, temporal, and spectral resolutions is rapidly developing [1] and several countries have established relatively adequate satellites for ground observation; these satellites include the Landsat series of satellites and the moderate-resolution imaging spectroradiometer (MODIS) sensors on



Citation: Chu, B.; Gao, F.; Chai, Y.; Liu, Y.; Yao, C.; Chen, J.; Wang, S.; Li, F.; Zhang, C. Large-Area Full-Coverage Remote Sensing Image Collection Filtering Algorithm for Individual Demands. *Sustainability* **2021**, *13*, 13475. https://doi.org/ 10.3390/su132313475

Academic Editors: Gavin McArdle, Mir Abolfazl Mostafavi and Hamidreza Rabiei-Dastierdi

Received: 14 October 2021 Accepted: 26 November 2021 Published: 6 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). board the Terra and Aqua satellites launched by the U.S.; the WorldView series of highresolution satellites; the Satellite pour l'Observation de la Terre (SPOT)-series satellites; Japan's Advanced Land Observing Satellite (ALOS); and China's GaoFen-series, HuanJingseries, and ZiYuan-series satellites. With the launch and use of various satellites, the amount of remote sensing data available has grown exponentially. For example, the GaoFen-2 satellite alone has collected more than two million scenes since its launch in 2014 [2]. Because of the multitude of remote sensing images, the development of efficient organization and storage methods, especially methods for retrieval, has become an urgent problem for remote sensing ground systems.

Remote sensing images are currently widely used in various applications for urban study and planning, such as road planning, real-time monitoring, and landcover survey. The general process is that users retrieve all image collections according to the delineated area, manually select the image subset that can fully cover the area of interest, and, with further processing, finally complete applications by mosaicking the images into a fullcoverage image [3,4]. However, with the explosive growth of remote sensing image data, the traditional manual image selection mode has difficulty meeting the current demands of tasks to select the optimal image collection from thousands or even hundreds of thousands of candidate images [5]. Additionally, according to different demands from different applications, users have different preferences for selecting image collections; these preferences may differ in terms of timeliness, cloudiness, spatial resolution, etc. For example, land and resource surveys of a very large urban require remote sensing images that have high coverage, but the spatial resolution of these images does not have to be very high [6]. However, in the task of urban detail monitoring, the spatial resolution becomes the most important factor. When faced with problems such as global mapping or extracting large impervious surfaces, image filtering based on image quality and coverage is usually an unavoidable step [7,8]. In addition, image filtering is important to remote sensing distribution websites, such as United States Geological Survey (USGS) Explorer, when the retrieval task involves many images. Detecting illegal buildings requires images to have relatively high timeliness and spatial resolution [9]. In summary, because of the multitude of remote sensing data, there is an urgent need to find a method for image collection filtering that can automatically and quickly extract a subset of remote sensing images that meet users' individual demands (concerning cloudiness, timeliness, and resolution) with few redundant images, while ensuring full area coverage.

In view of the above problems, this paper proposes a large-area full-coverage remote sensing image collection filtering algorithm for individual demands (LFCF-ID). Under the premise of full coverage through an image collection containing a minimum number of images, automatic filtering is accomplished by optimizing a genetic algorithm (GA) according to different users' needs for image quality, area coverage, image timeliness, etc. Additionally, this method maximizes the offline loop computing processes and designs multiple optimization methods, thereby greatly improving the algorithm's operating speed.

The main contributions of this paper are as follows:

We propose a new mode for remote sensing image collection filtering that aims to tackle the difficult problems faced by different urban researchers and planners when searching for image collections that fully cover the area of interest while considering different demands for spatial resolution, timeliness and image quality.

Focusing on the above mode, we propose the LFCF-ID, which can fully cover the interest area by using the smallest number of remote sensing images while satisfying users' demands concerning spatial resolution, timeliness, and image quality.

By using a greedy algorithm, we conduct a series of contrast experiments that focus on different study areas. The experimental results show that the proposed method is robust and, compared to existing methods, obtains better filtering results. Additionally, this method has great speed advantages over the manual filtering method.

2. Related Work

Current research on image filtering focuses mainly on retrieving image slices from massive images that are similar in land cover to input images; this research includes studies by Akshara [10], Liu [11], and Li [12]. The fundamental purpose of the abovementioned remote sensing image retrieval work was to analyze and retrieve images containing similar scenes. However, the above methods cannot solve the problem presented by users that filter images according to coverage, resolution, timeliness, etc.

Other scholars have conducted some research on remote sensing data filtering based on image attributes. Li proposed an image pyramid-oriented spatial indexing algorithm based on a linear quadtree; this algorithm improved the efficiency of coding and retrieving single-view images [13]. Xie designed a remote sensing image catalog data description framework and, based on this framework, designed an efficient catalog data storage organization scheme and positioning retrieval, qualitative retrieval, and combined retrieval algorithms to solve the problem of achieving fast and accurate image localization under distributed storage [14]. However, the studies by these scholars focused on parallelism and indexing mainly at the data level. Area coverage and filtering of image collections to meet individual demands still require manual intervention. There is still no solution for meeting these key issues of full coverage and the satisfaction of demands. Egenhofer studied the sketch spatial data retrieval method [15]; Lee proposed a visual query based on topological relationships in GIS [16]; and Shekhar, Volker, Gaede, and others studied information retrieval related to spatial information [17,18].

To date, few studies have focused on satisfying both full coverage and individual demands in remote sensing data filtering. He proposed a single-phase full-coverage filtering algorithm for remote sensing images [19]. Although this algorithm could complete full coverage of a given area, the algorithm did not consider factors such as cloudiness and resolution. Thus, it was difficult to obtain satisfactory results using this algorithm. Zuo started from remote sensing tiled data and conducted data filtering in units of tiles. A full-coverage retrieval model was designed, but this model still required manual review and interactive filtering of the results, which could not avoid the problem of low efficiency when the search area or the amount of data was large [5]. Li proposed a data set filtering model for optimal remote sensing image area coverage [20]. This model normalized the user-defined cloud range, time range, sensor type, and resolution. The score of each image on a regular grid was calculated, and the image with the highest score on each grid was selected as the filtering result. The filtering results of this model did not consider the image overlap ratio, so the filtered image collections were very repetitive.

Currently, the public can obtain remote sensing images for free from some websites (e.g., USGS Earth Explorer, the European Space Agency (ESA)'s sentinel mission, and the National Aeronautics and Space Administration (NASA)'s Reverb); these websites can also simply filter images according to cloud cover, latitude and longitude, and other conditions. However, unlike the proposed method, inputting cloud cover and latitude and longitude in the website for retrieval will enable hundreds, or even more, qualified images to be obtained, and images will greatly overlap. Users who want to achieve full coverage of very large areas by selecting multiple remote sensing images still cannot avoid the problem of selecting spatially complementary images from many images that the website filters. Even if several complementary images are selected to fully cover the area, there is no guarantee that, compared with other highly overlapping images, the image at this time is of relatively good quality.

In summary, there is currently no mature image collection filtering method that can both achieve full-coverage and satisfy the different demands of users.

3. Proposed Method

The workflow of the LFCF-ID proposed in this study is shown in Figure 1. This framework includes four main steps: obtaining the image collection to be filtered; fast, coarse filtering of the image collection; executing a genetic filtering algorithm; and result

images.

optimization. Generally, attributes such as cloudiness, location, and time vary greatly among remote sensing images. Most of the images that are available cannot satisfy users' specific demands. Thus, in the first step, it is necessary to obtain an image collection that is worth filtering by removing many redundant images according to the user's specific preset area, time interval, and minimum thresholds for cloudiness and resolution. The output is used as the image collection to be filtered to ensure that any image in the collection can meet the basic needs of users. Then, fast, coarse filtering of the image collection is performed. In this step, a score is designed, mainly in grid units, that can represent how well the image collections satisfy the user's demands. Images with the top k scores in each grid are selected as the result of coarse filtering. Next, the GA is used to further filter the coarse filtering results according to the fitness score, which can provide a comprehensive expression of the user's demand satisfaction and coverage, and the optimal full-coverage image collection result filtered to meet the individual demands is obtained. Finally, the final image collection result is obtained by result optimization, which is conducted to slightly change the optimal full-coverage image collection result by adding or subtracting a few

The blue area in Figure 1 is the area preset by the user, and the yellow rectangles are the boundaries of each remote sensing image. The figure shows that with the implementation of each step, the filtered image collection becomes increasingly optimal.



The display on the map for the output of each step

Figure 1. Workflow for the LFCF-ID.

Considering that the image collection filtering algorithm needs to meet the individual demands of different users, mathematical modeling of the requirements is essential. The user's demands are reflected mainly in terms of cloudiness, resolution, and timeliness, and this paper uses a weighted triplet to model the user demands.

$$Model_{user} = \left\{ W_{gsd}, W_{cloud}, W_{time} \right\}$$
(1)

 W_{gsd} , W_{cloud} , and W_{time} are the input weights preset by the user. W_{gsd} indicates the user's demand weight for the spatial resolution of the image. A larger value of W_{gsd} indicates that the user prefers images with a higher resolution in the filtering result. W_{cloud} indicates the user's demand for image cloudiness. A larger value of W_{cloud} indicates that the user prefers images with fewer clouds in the filtering result. The weight of W_{time} indicates the user's demand for the timeliness of the image. The larger the value, the more the user hopes to obtain images in the filtering results that are close to the deadline selected by the user themselves.

3.1. Obtaining the Image Collection to Be Filtered

In this method, we first need to predefine several required parameters in the filtering workflow, as shown in Table 1:

Table 1. Definition of the preset parameters.

Parameter	Definition
Target_Area	Specific area preset by users. The goal of the LFCF-ID is to filter and obtain the full-coverage image collection of this area
T_{start}, T_{end}	The time interval is preset by users to obtain the images to be filtered
GSD_{\min}	The lowest image resolution that the user can accept in the filter result
<i>Cloud</i> _{max}	The highest image cloudiness that the user can accept in the filtered result
Coverage _{min}	The lowest coverage that the user can accept
W_{gsd}	The user's demand weight for the spatial resolution of the image
W _{cloud}	The user's demand weight for the image cloudiness
W_{time}	The user's demand weight for the timeliness of the image

In this step, all the images available are used as the input data, which can be thousands of scenes. We obtain the images that can satisfy the preset parameters Target_Area, T_{start} , T_{end} , GSD_{min} , and $Cloud_{max}$ as the initial image collection, which will be filtered in the next steps. The initial image collection to be filtered is denoted by $I_ori = \{I_1, I_2, ...\}$. I_ori is used as the input of the fast, coarse filtering step, which performs a further filtering operation according to W_{gsd} , W_{cloud} , and W_{time} .

3.2. Fast, Coarse Filtering of the Image Collection

In this step, we conduct a coarse filtering of I_ori by designing a series of standard scores consisting of W_{gsd} , W_{cloud} , and W_{time} . A subset of $I_ori = \{I_1, I_2, ...\}$ can be obtained as the output, which will be used as the input of the GA in the next step.

3.2.1. Creation of Global Grids

In this paper, coarse filtering and genetic filtering algorithms need to calculate several scores to rank images by using the grid as the computing unit. Large grids may cause fineness errors due to excessively large benchmarks, and small grids will be too fragmented and thus can result in many calculations, so the grid size should be as reasonable as possible. In the experiment, we divide the grid according to different standards such as $0.01^{\circ} \times 0.01^{\circ}$, $0.05^{\circ} \times 0.05^{\circ}$, $0.1^{\circ} \times 0.1^{\circ}$, $0.3^{\circ} \times 0.3^{\circ}$, and $0.5^{\circ} \times 0.5^{\circ}$. We find that among these standards, $0.1^{\circ} \times 0.1^{\circ}$ works best. Therefore, we use $0.1^{\circ} \times 0.1^{\circ}$ in all experiments.

3.2.2. Coarse Filtering Based on Grids

First, we calculate the grids denoted by $g = \{g_1, g_2, \dots, g_i, \dots, g_n\}(i = 1, 2, \dots, n)$, which can cover the Target_Area preset by the user. Then, we search for the images denoted by $I_ori_i = \{I_ori \cap g_i\}$, which cover g_i in I_ori . Because I_ori_i contains many images, only a few images belong to the optimal image collection that meets the user's demands. It is inappropriate to use all images in I_ori as the input of the GA, as doing so can cause data redundancy. Data redundancy may make it difficult for the GA to converge in the next step and also reduces the calculation efficiency of the GA. Thus, to reduce the number of images to be filtered by the GA, an image scoring formula S_I is designed to calculate the satisfaction of the user's demands by each image in this paper. S_I is calculated as

$$S_I = W_{gsd} \times S_{gsd} + W_{cloud} \times S_{cloud} + W_{time} \times S_{time}$$
(2)

$$W_{gsd} + W_{cloud} + W_{time} = 1, \ 0 \le W_{gsd} \le 1, \ 0 \le W_{cloud} \le 1, \ 0 \le W_{time} \le 1$$
 (3)

where S_{gsd} is the resolution score of the remote sensing image, S_{cloud} is the cloudiness score of the remote sensing image, and S_{time} is the timeliness score of the remote sensing image. S_{gsd} is calculated as follows:

$$S_{gsd} = \frac{gsd_{\max} - gsd_{present}}{gsd_{\max} - gsd_{\min}},$$
(4)

where gsd_{max} is the highest resolution of all images in I_{ori} , gsd_{min} is the lowest resolution of all images in I_{ori} , and $gsd_{present}$ is the resolution of the image to be calculated.

 S_{cloud} is calculated as follows:

$$S_{cloud} = \frac{C_{\max} - C_{present}}{C_{\max} - C_{\min}},$$
(5)

where C_{max} represents the theoretical maximum value of the image cloudiness (this value is actually 100%), C_{min} represents the theoretical minimum value of the image cloudiness (this value is actually 0%), and $C_{present}$ represents the cloudiness of the image to be calculated.

S_{time} is calculated as follows:

$$S_{time} = \frac{T_{present} - T_{start}}{T_{end} - T_{start}}$$
(6)

where T_{start} and T_{end} are defined in Table 1 and $T_{present}$ represents the image shooting, which is calculated in days.

The formula for S_I can be used to calculate the score of each image under the user's demands, which are modeled by $Model_{user} = \{W_{gsd}, W_{cloud}, W_{time}\}$. The best k images are selected as the coarse filtering result of each grid g_i by ranking the S_I values of the images in I_ori_i ; the best k images of all grids are then merged together as the final coarse filtering result of the Target_Area; this result is denoted by $I = \{I_1, I_2, \dots, I_m\}, m \le n * k$, where n is the number of grids that can fully cover the Target_Area.

The coarse filtering method above can quickly remove many images, thereby producing a result that can satisfy the preset parameters Target_Area, T_{start} , T_{end} , GSD_{min} , and $Cloud_{max}$, but is not suitable enough for the final result of the LFCF-ID. After coarse filtering, the number of images in the image collection is reduced to the same magnitude as the number of grids n, thus possibly greatly improving the efficiency of the GA algorithm.

3.3. Further Filtering by the Genetic Algorithm

The ultimate purpose of the filtering task in this paper is to obtain an image collection containing the fewest images that can exactly fully cover the Target_Area specified by the user while best satisfying the user's demands. Since each image usually covers more than one grid, we select the k best images in each grid as the output of the coarse filtering;

this selection will result in a high level of overlap between images. Many images are unnecessary and the image collection is still redundant. Therefore, it is time to find an optimization method to select the best combination of images as the optimal image collection that can meet the full-coverage requirement and the lowest repeated coverage requirement.

As a well-known search and optimization method, the GA has been successfully applied to intelligent optimization problems in various remote sensing applications, such as image classification [21–23], image segmentation [24], feature extraction [25–27], and quantitative inversion [28,29]. The quasi code for a GA is Algorithm 1.

Algorithm 1 Genetic Algorithm.
Input: Pc: possibility of cross
Pm: possibility of mutation
m: the number of genes in a population
Output: optimal population
1: initialize population including m genes
2: calculate the fitness score for each gene
3: repeat
4: select m genes from population by using the roulette method
5: $if(random(0,1) < Pc)$
{
select two genes randomly
cross between the two selected genes
}
6: $if(random(0,1) < Pm)$
{
select one gene randomly
mutation for this selected gene
}
7: calculate fitness score for each gene
8: until(reaches stop condition)

Inspired by the GA, this study uses a grid as the calculation unit to calculate the fitness score, which can model the image collection's state of coverage and satisfaction of personalized demands. The GA is used to continuously optimize the fitness score to obtain the optimal filtering results. In this algorithm, the input form and fitness functions should be defined according to the problem. The unit of the input is called the chromosome in the GA; this input can be a solution to the problem. In this paper, the chromosome refers to an image collection. The chromosome is composed of many genes that can be represented in binary, and refers to the images in the image collection. At the beginning of the GA, a population is formed by randomly generating multiple possible chromosomes. Biological selection, crossover, and mutation operations of the population are used to simulate the biological genetic evolution and achieve the next-generation population according to a preset fitness function. The higher the chromosome's fitness score, which is calculated by the fitness function, the higher the probability of saving the chromosome for the next generation. Therefore, the fitness score of the chromosome can improve generation-bygeneration. Eventually, the chromosome with the best fitness score is obtained as the final optimization result.

3.3.1. Population Initialization

First, we need to define the meaning of a chromosome in the GA according to the problem to be solved in this paper and generate an initial population. Since the final result of the LFCF-ID is an image collection, we define the chromosome as an image collection, which is denoted by G_i .

$$G_i = \{A_1, A_2, \dots, A_j, \dots, A_m\} \ (A_i = 0 \text{ or } A_i = 1)$$
(7)

The variable m, which denotes the number of images in the coarse filtering result, is set to the length of G_j . The h-th gene $A_h = 0$ indicates that image collection G_j does not include the *h*-th image in the coarse filtering result, and $A_h = 1$ indicates that image collection G_j includes the h-th image in the coarse filtering result. The coding method for an image collection is shown in Figure 2.

A_1	A ₂	A_3	A_4	•••	$A_{ m j}$	•••	$A_{\mathrm{m-l}}$	$A_{ m m}$
1	1	0	1	•••	0		1	0
							1	

 $G_i = \{1, 1, 0, 1, \dots, 0, \dots, 0\}$ the collection only contains $\{A_1, A_2, A_4, \dots, A_{m-1}\}$

Figure 2. Coding method for an image collection.

After randomly generating multiple binary codes of length m, we can form the initialization population. In this paper, we set *P*, which denotes the number of image collections in the population, to an empirical value of 10.

$$P = \{G_1, G_2, \dots, G_{10}\},\tag{8}$$

3.3.2. Fitness Score Function

The second important process of the GA is to design a reasonable fitness function to describe the satisfaction of each image collection G_j in terms of coverage and user demands, which are denoted by S_{G_j} . In this paper, we use a grid as the computing unit to calculate the fitness score of the image collection.

To obtain S_{G_j} , we first need to define S_{g_i} , which denotes the fitness function of the image collection on each grid g_i . Since usually more than one image can cover grid g_i in G_j in most cases, we calculate the S_I (Equation (2)) of all images that can cover grid g_i in G_j and take the highest score as S_{g_i} . The calculation formula for S_{g_i} is as follows:

$$S_{g_i} = \max_{I_i \cap g_j \neq \varnothing} S_{I_i} \times A_i \tag{9}$$

where $I_j \cap g_i \neq \emptyset$ indicates that the intersection of the *j*-th image and the *i*-th grid is not empty. Then, we need to define the fitness function S_{G_j} of the image collection. Considering that the fitness score is related to more than the satisfaction of the user's demands, the coverage also has a certain impact. We define the parameter *OL* to represent the coverage of image collection G_j over the Target_Area. *OL* is calculated as follows:

$$OL = \begin{cases} cr & cr < 1\\ \sum\limits_{\substack{i \in rids \\ -\frac{i=1}{n_{Grids}}} ct_i & , cr = \frac{n_{GridsOfCovered}}{n_{Grids}} \end{cases}$$
(10)

where *cr* denotes the coverage rate, $n_{GridsOfCovered}$ denotes the number of grids covered by the image, ct_i denotes the number of times the *i*-th grid is covered by the image, and n_{Grids} denotes the number of grids in the administrative area.

A value of less than 1 for *OL* indicates that the Target_Area cannot be completely covered by G_j ; a value of 1 for *OL* indicates that the Target_Area is completely covered by G_j ; and a value larger than 1 for *OL* indicates the number of layers that repeatedly cover the Target_Area. Commonsensically, when *OL* is equal to 1, G_j can cover only the Target_Area without redundancy, and S_{G_j} should be the largest at this time. When *OL* is greater than 1, S_{G_j} will decrease by a multiple. When *OL* is less than 1, G_j cannot completely

cover the Target_Area; users cannot tolerate the lack of coverage. Considering the above characteristics, S_{G_i} is calculated as follows:

$$S_{G} = \begin{cases} \left(\sum_{i=1}^{n} S_{g_{i}}\right) \times \frac{1}{OL} + 1 & if(OL \ge 1) \\ 2^{-\frac{1}{OL}} & if(OL < 1) \end{cases},$$
(11)

The above formula shows that when OL = 1, S_{G_j} is equal to adding 1 to the scores of all grids S_{g_i} . Bias 1 is added to ensure that the score for full coverage is greater than the score for incomplete coverage.

After the population is initialized by the GA, reproduction, crossover, and mutation are carried out to simulate the inheritance process to obtain the best gene.

We iterate the selection, crossover, and mutation steps until convergence is reached. The condition for convergence is set as the highest fitness score of the image collection not changing for fifty iterations. In Figure 3, the red line means that although the preset threshold is not reached, as the generation increases, the fitness score changes by less than 0.1 after 50 iterations. This means that the algorithm has converged and the current optimal image collection can be the final result.



Figure 3. Termination condition for the GA.

3.4. Filtering Result Optimization

Through the above steps, the optimal image collection in the last population can be obtained; this collection is denoted by G_{best} . Although the GA can guarantee evolution to a better population, there are some small flaws, which are generally caused by one or two redundant images. Therefore, we need to optimize the structure of G_{best} . The optimization involves traversing each gene A_j in G_{best} . When $A_j = 1$, A_j will be set to 0 to generate a new chromosome $G_{best_{new}}$. If $S_{G_{best}}$ is then increased, the new chromosome $G_{best_{new}}$ will be replaced with G_{best} until there is no further increase in $S_{G_{best}}$. G_{best} is then the final image collection filtering result.

4. Implementation and Performance Analysis

4.1. Experimental Region

Different levels of government urban planning departments need to observe regions of different scales. Provincial governments need to observe and conduct spatial statistical

analysis on the whole province, and urban planning departments at the city level are concerned with whole cities. In order to verify the accuracy and efficiency of the LFCF-ID for different levels of administrative regions, the experiment used multiple administrative regions in China as experimental regions, including Shijiazhuang (medium city), Beijing (large city), and Hebei (province), which are shown in Figure 4.



Figure 4. Experimental region in this study.

- 4.2. Data
- 4.2.1. Data Source

To effectively verify the ability of the LFCF-ID to filter image collections, the data used in the experiment should be as diverse as possible. Therefore, data from multiple satellites was selected; the main satellites selected include GaoFen, ZiYuan, and Jilin, which use multispectral, synthetic-aperture radar (SAR), hyperspectral, and other sensor types with spatial resolutions from sub-meter to hundreds of meters and widths from 5 km to 720 km. Detailed information is shown in Table 2.

Table 2. Information on the images used.

Name of Satellite	Sensor Type	Resolution	Width
GaoFen1	Multispectral	2/8 m	60 km
GaoFen2	Multispectral	1/4 m	45 km
GaoFen3	SÂR	1–500 m	5–650 km
GaoFen4	Multispectral/infrared	50/400 m	400 km
ZiYuan3-02C	Multispectral	5/10 m	51 km
ZiYuan3	Multispectral	2.1 m	51 km
ZiYuan-CB04	Multispectral	2.36 m	113 km
Huanjing-1A	Multispectral/hyperspectral	30/100 m	360/50 km
Huanjing-1B	Multispectral/infrared	30/300 m	360/720 km
TRIPLESAT1	Multispectral	0.8/3.2 m	51 km
TRIPLESAT2	Multispectral	0.8 m/3.2 m	51 km
TRIPLESAT3	Multispectral	0.8 m/3.2 m	51 km
LANDSAT8-L1TP	Multispectral/infrared	15/30/100 m	185 km



In this paper, we selected 111,414 images from the above satellites within the time range of 2008 to 2018. We calculated the statistics of the data from the satellites and sensors; these statistics are shown in Figure 5.

Figure 5. Statistics of the number of images for different satellites and sensors.

4.2.2. Image Data Distribution of Different Types and Regions

To more intuitively describe the distribution of data, we statistically distributed the data in three aspects: time, resolution, and cloudiness. Figure 6a shows the statistics of the images in terms of four levels: cloudless, low cloudiness, medium cloudiness, and high cloudiness. The number of cloudless images is the largest, and the number of images decreases with increasing cloudiness. Figure 6b shows the statistics of the number of images at different spatial resolutions. Although there is a certain difference in the amount of data in each section, there are enough images in each section. Figure 6c shows the statistics of the amount of image data for each year from 2008 to 2018 in units of years. This image collection covers all years and is concentrated from 2015–2017. To verify the applicability of the proposed method in different regions, this paper selected three areas of different sizes as experimental areas: Shijiazhuang, Beijing, and Hebei; additionally, we statistically analyzed the data distributions of different areas. Figure 6d shows that the number of images is proportional to the area of the region, and the image data for each area are sufficient. In summary, the distribution of the data used in the experiment in different types and regions is sufficient and is thus able to support the verification of this study.





Figure 6. (a) Statistics on cloudiness, (b) Statistics on resolution, (c) Statistics on timeliness, and (d) Statistics on study area.

4.3. Experiment and Analysis

60,000

e 50,000

40,00 30,000

5 20,000

To fully verify the accuracy, robustness, and efficiency of the proposed method, multiple experiments were designed from different perspectives. First, the top k images were selected for each grid during coarse filtering; this selection can directly affect the efficiency of the algorithm by affecting the length of the chromosome. This paper compared experiments with different k values. Then, under the optimal k value, experiments were performed in different regions, with different numbers of images to be filtered, and different weights of users' demands. Designing experiments using different regions can verify the robustness of the algorithm. Experiments using different numbers of images to be filtered can directly verify the speed of the algorithm. Designing experiments with different demands can verify whether the proposed method can meet the users' individual demands. In addition, this article also designed experiments to compare this method with manual methods and other algorithms to verify the superiority of the proposed method.

4.3.1. Experiments in Various Situations

Experimental results for different k values a.

We take the data from Shijiazhuang in 2018 as the input and set $\{W_{gsd}, W_{cloud}, W_{time}\}$ to {0.333, 0.333, 0.333}. We set the k value from 1 to 6. The experimental results are shown in Figure 7 and Table 3. The black box in Figure 7 indicates the boundary of each remote sensing image.

k=3 k=2 k=1 k=6

Figure 7. Filtering results with different k values.

k=4

Table 3. Information on the filtering results with different k values.

k	Length of G_i	Fitness Score of G _{best}	Image Number of G _{best}
1	29	1.661959	19
2	56	1.687312	18
3	75	1.732062	17
4	101	1.728173	19
5	133	1.716126	20
6	163	1.691373	21

k=5

Figure 7 shows that regardless of the k value, the full-coverage requirements of the filtering results are basically met, but the composition of the image collection is different. Table 3 shows that when k = 1, $S_{G_{best}}$ can reach only 1.661959, which is far lower than the results obtained under other k values. This result occurs because the coarse filtering in the algorithm when k = 1 can be understood as equivalent to the greedy solution, which roughly finds a single image that best meets the user's demands for each grid, and the filtering results can be obtained by combining the results of all grids. Since the size of each remote sensing image is larger than the grid, greedy conditions will make the OL of the image collection very large; this can result in a low fitness score.

When k gradually increases from 1 to 3, the total score continues to increase, and the image number of G_{best} gradually decreases. This result indicates that better image collection can be achieved when the number of images participating in the GA gradually increases.

When k continues to increase from 3 to 6, the score gradually decreases. This result indicates that when the value of k increases to a certain level, the quality of the newly added images will decrease compared to when k is small, and the improvement is not obvious. When k further increases, the length of G_i will also increase exponentially, thus affecting the results of the GA by leading to a local optimum and making it difficult to find the optimal solution. For example, when k = 6 in Table 3, the length of G_i reaches 163 and $S_{G_{hest}}$ is reduced to 1.691373.

b. Experimental results for different regions

To verify that the proposed method has robustness for different areas with irregular shapes, this study performed experiments in Shijiazhuang, Beijing, and Hebei within a fixed time interval under the premise of setting $\{W_{gsd}, W_{cloud}, W_{time}\}$ to {0.333, 0.333, 0.333}. The filtering results are shown in Figure 8.



Figure 8. Filtering results in different study areas.

Figure 8 shows that when the proposed method is used for filtering in different regions, although the area and shape may differ greatly, good coverage can be obtained for all regions. Table 4 shows that the *OL* values of the image collection after filtering in different regions are 1.320225, 1.243119, and 1.432177, all of which are below 2, and there are no repeated overlays caused by image redundancy.

Study Area	Area (km²)	rea (km ²) Coverage of G_{best} Image Number of G_{best}		OL of G _{best}	Time Consumption (s)	
Shijiazhuang	15,850	100%	23	1.320225	3.1044	
Beijing	16,410	100%	22	1.243119	3.8220	
Hebei	188,900	100%	21	1.432177	68.6900	

Table 4. Information on the filtering results in different study areas.

c. Experimental results for algorithm robustness verification

To verify the robustness of our algorithm, we performed experiments with 50 iterations in Beijing, Hebei, and Shijiazhuang under the same input (k set to 3, the cloud weight set to 0.3, the timeliness weight set to 0.35, and the resolution weight set to 0.35). We produced two boxplots of fitness scores and OL. The boxplot is a statistical graph used to display the dispersion of a dataset and can therefore be used to verify the robustness of the algorithm. The boxplots are shown in Figure 9.

In the boxplot, the middle line of the box is the median of the data and thus represents the average level of the sample data. The upper and lower limits of the box are the upper and lower quartiles of the data, thus indicating that the box contains 50% of the data, so the broadband of the box reflects the degree of data fluctuation to a certain extent. There are additional lines above and below the box; these lines represent the maximum and minimum values. Sometimes, some points are outside of the upper and lower limits, and these points are outliers. As shown by the boxplots of the fitness scores, the distance between the upper quartile and the lower quartile of Hebei is within 0.07, while the distances of Beijing and Shijiazhuang are within 0.05, thus indicating that the fluctuation of the fitness score in the 50 experiments was very small. As the *OL* boxplot shows, the distance between the upper quartile and the lower quartile of Hebei is within 0.18, while the distances of Beijing and Shijiazhuang are within 0.1, thus indicating that the fluctuation of *OL* in the 50 experiments was very small. The distance between the maximum and minimum values in the two boxplots is also within the acceptable range, and there are no outliers. In summary, this algorithm is very robust.



Figure 9. The boxplots of fitness score and OL.

d. Experimental results for different image numbers of *I_ori*

To verify the efficiency of this method when processing images of different orders of magnitude, we performed experiments under the premise of fixing other conditions and controlling the number of images processed by setting time intervals of different lengths. The results are shown in Figure 10 and Table 5.



Figure 10. (a) Filtering effect of tens of thousands of pictures, (b) Filtering effect of thousands of pictures, (c) Filtering effect of hundreds of pictures.

Figure 10 shows that the proposed method exhibits good filtering results for input images of all orders of magnitude. In detail, Table 5 shows that with the gradual increase in the amount of I_{ori} , the OL of I_{ori} increases from 14.93578 to 500.2661, and the OL of G_{best} decreases to approximately 2. The fitness score of G_{best} increases steadily from 1.5579 to 1.750479. This result shows that the proposed method can obtain better results regardless of whether the amount of data is in the hundreds, thousands, or tens of thousands. An increasing amount of input data provides more choices for the filtering process and produces better results. Table 5 shows that the time consumed by the proposed method remains stable when the order of magnitude of the input images increases; the time consumed by traditional methods should increase exponentially.

Image Number of I_ori	OL of I_ori	Fitness Score of G _{best}	OL of G _{best}	Time Consumption (s)
243	14.93578	1.557944	1.633028	2.886000156402588
486	20	1.566519	1.394495	7.019999980926514
626	35.78899	1.584082	1.426606	3.6347999572753906
710	39.62844	1.705311	1.353211	3.5411999225616455
867	45.72477	1.600818	1.490826	4.648799896240234
1202	60.09174	1.60953	1.522936	2.683199882507324
1928	97.96789	1.682792	1.385321	2.8703999519348145
2982	146.2477	1.792568	1.224771	3.75959992408752
4094	198.2018	1.691633	1.408257	5.50679993629455
4955	240.4128	1.689813	1.40367	4.040399789810181
5975	289.156	1.735492	1.325688	5.756400108337402
6935	333.2982	1.753975	1.261468	5.30400013923645
7994	379.8945	1.762623	1.256881	4.055999994277954
9019	426.5505	1.715952	1.348624	6.910799980163574
10008	500.2661	1.750479	1.307339	8.252399921417236

Table 5. Detailed data on the filtered results for different orders of magnitude.

e. Experimental results for individual demands

Different types of urban study and planning tasks have different demands for remote sensing images. For example, in the task of urban road planning, planners need to master the distribution of all roads in the city, which requires high resolution of remote sensing images. In the urban real-time monitoring task, it is necessary to carry out high-frequency real-time monitoring of key areas such as illegal buildings in the urban area, and high timeliness of remote sensing images is required. In order to effectively verify whether the proposed method can meet the demands of urban study and planning tasks, multiple experiments were designed from high to low in terms of timeliness, resolution, and cloudiness.

To quantitatively evaluate the cloudiness, resolution, and timeliness of the image collection, three measures are defined: the average grid cloudiness is denoted by Aver_Cloud, the average grid resolution is denoted by Aver_GSD, and the average grid timeliness is denoted by Aver_Time. We obtain the g_i cloudiness score, which is denoted by $S_{cloud_{g_i}}$; the g_i resolution score, which is denoted by $S_{gsd_{g_i}}$; and the g_i timeliness score, which is denoted by calculating S_{gsd} (Equation (4)), S_{cloud} (Equation (5)), and S_{time} (Equation (6)) of the image with the highest score in grid g_i . We obtain Aver_Cloud, Aver_GSD, and Aver_Time by calculating the average $S_{cloud_{g_i}}$, $S_{gsd_{g_i}}$, and $S_{time_{g_i}}$ for all the grids. The formulas are as follows:

Aver_Cloud =
$$\frac{1}{n} \sum_{i=1}^{n} S_{cloud_{g_i}}$$
, (12)

Aver_Time =
$$\frac{1}{n} \sum_{i=1}^{n} S_{\text{time}_{g_i}}$$
 (13)

$$Aver_GSD = \frac{1}{n} \sum_{i=1}^{n} S_{gsd_{g_i}},$$
(14)

where *n* is the number of grids covered by the Target_Area.

In Figure 11, W_{gsd} in the second line of images gradually decreases from left to right from 1 to 0.1. The color of the images in the filtered image collection gradually changes from all green to partially red and partially pink until blue appears, thus indicating that the image resolution in the filtered results is gradually decreasing. In the task of urban road planning, planners can set W_{gsd} to a larger value such as the first figure in the second row. Planners can obtain an image collection with a resolution better than 3 m to achieve full coverage of the Shijiazhuang area in which the road and other small-scale landcover can be seen.



Figure 11. Image collection filtering results obtained under different weights.

 W_{time} in the third line of the image gradually decreases from left to right from 1 to 0.1. This figure shows that the color of the images in the filtered image collection gradually changes from all green to partially red, and from having little blue to partial blue appearing, thus indicating that the time of the images in the filtered result is far from the last date. In the task of urban real-time monitoring, people can set W_{time} to a larger value such as the first figure in the third row. Then, an image collection within 10 days can be obtained to observe buildings and other illegal landcover.

 W_{cloud} in the first row of images gradually decreases from left to right from 1 to 0.1. The color of the images in the filtered image collection gradually changes from all green to partially red and partially pink until blue appears, thus indicating that the cloudiness of the images in the filtered result is gradually increasing. For the task of a large-scale landcover survey, people can set W_{cloud} to a larger value such as the first figure in the first row. Then, an image collection without any clouds can be obtained to avoid situations in which land is blocked.

Table 6 provides detailed information for each image in Figure 12. Figure 12 is a visualization of the changes in Aver_Cloud, Aver_GSD, and Aver_Time according to $\{W_{gsd}, W_{cloud}, W_{time}\}$; these changes are also shown in Table 6. As shown in Figure 12a, as the cloudiness increases, the Aver_Cloud score gradually increases from 0.965056 to 1, and the scores of Aver_GSD and Aver_Time gradually decrease from 0.961545 and 0.946545 to 0.708643 and 0.225515, respectively. In Figure 12 b, as the grid resolution increases, the Aver_GSD score gradually increases from 0.737714 to 0.986794, and the Aver_Cloud and Aver_GSD scores gradually decrease from 0.999555 and 0.986116 to 0.958371 and 0.675474, respectively. In Figure 12 c, as time increases, the Aver_GSD scores gradually decrease from 0.999555 and 0.986116 to 0.985105, and the Aver_Cloud and Aver_GSD scores gradually decrease from 0.99911, respectively. In summary, the LFCF-ID can filter image collections according to preset weights to meet individual demands.

4.3.2. Comparison with Other Methods

There is currently no other study similar to this paper, so the algorithm in this paper is only compared with manual filtering and greedy methods. The manual filtering method refers to the process by which a person selects from several images to obtain the results that he or she feels meet the demands. The greedy method refers to traversing all the images of each grid to obtain the best image and combining the best images of all the grids as the filtering result. The experimental results are shown in Table 7.

W _{cloud}	W_{gsd}	W _{time}	Fitness Score	Aver_Cloud	Aver_GSD	Aver_Time
1	0	0	1.67424	1	0.708643	0.225515
0.6	0.2	0.2	1.70223	0.995618	0.95063	0.915011
0.3	0.35	0.35	1.7221	0.980618	0.960834	0.934039
0.1	0.45	0.45	1.67466	0.965056	0.961545	0.946545
0	1	0	1.73802	0.958371	0.986794	0.675474
0.2	0.6	0.2	1.69685	0.983989	0.97199	0.915378
0.35	0.3	0.35	1.68953	0.979607	0.95591	0.945887
0.45	0.1	0.45	1.66885	0.995955	0.737714	0.986116
0	0	1	1.67702	0.99764	0.629191	0.985105
0.2	0.2	0.6	1.74005	0.988258	0.775825	0.971083
0.35	0.35	0.3	1.76462	0.999101	0.964984	0.915991
0.45	0.45	0.1	1.70676	0.999494	0.977812	0.869792

Table 6. Information on the filtering results with different weights.



Figure 12. Changes in Aver_Cloud, Aver_GSD, and Aver_Time according to different weight values. (a) Changes in Aver_Cloud according to W_{cloud} . (b) Changes in Aver_GSD according to W_{gsd} . (c) Changes in Aver_Time according to W_{time} .

 Table 7. Comparison of the time consumption and fitness scores of different methods.

Manual Filtering Method		Greedy Method		LFCF-ID $(k = 1)$		LFCF-ID (k = 3)		
Study Area	Time Consumption (s)	Fitness Score	Time Consumption (s)	Fitness Score	Time Consumption (s)	Fitness Score	Time Consumption (s)	Fitness Score
Shijiazhuang	48.986	1.7021	0.3432	1.4726	1.9032	1.6988	3.1044	1.7218
Beijing Hebei	100.738 652.584	1.6818 1.6492	$0.5460 \\ 8.0340$	1.4611 1.3015	2.4336 30.1704	1.7093 1.6176	3.8220 68.6900	1.7315 1.6708

Table 7 shows that in terms of time consumption, the greedy LFCF-ID (k = 1) and the LFCF-ID (k = 3) outperform the manual method. The greedy algorithm lacks optimization of the filtering results and directly uses the results of the grid retrieval as the output, so this algorithm takes the least time. However, the fitness score of the greedy method is poor compared to the fitness scores of the manual, LFCF-ID (k = 1), and LFCF-ID (k = 3) methods, thus indicating that the filtering effect of the greedy method is very poor. The LFCF-ID (k = 1) can be considered genetic optimization based on greedy retrieval. The LFCF-ID (k = 1) is slower than the greedy algorithm, but the fitness score is greatly improved. Compared with the LFCF-ID (k = 1), the LFCF-ID (k = 3) allows more images to be subsequently added to genetic optimization. Due to the increase in gene length, the time is slightly increased, and the fitness score is higher than that of other methods and is even better than that of the manual method. In summary, the comparison indicates that the LFCF-ID is 10 times faster than the manual method and exhibits a filtering effect that is better than the filtering effects of all existing methods.

5. Conclusions

This paper proposed a new remote sensing image filtering method that can support people to maximize their preferences in urban observation for study and planning tasks, on the premise of ensuring full coverage of a region. We first designed a coarse filtering strategy to reduce the dimensionality of the data; this reduction can save the most useful images and save time for subsequent optimization calculations. Then, a grid was used as the basic unit to calculate the fitness score of the image collections in terms of resolution, cloudiness, and coverage; the score was used as the index to evaluate the performance of the image filtering algorithm. Finally, the evaluation index was optimized by genetic iteration to optimize the image collections. We designed different sets of experiments to evaluate the performance of the LFCF-ID; the experimental results showed that the LFCF-ID could quickly achieve full-coverage filtering of images and simultaneously minimize repeated coverage as much as possible with different regions, different data volumes, and different demand preferences. The LFCF-ID showed great potential in solving how to automatically and quickly obtain full-coverage image collections of areas that meet the demands of preferences, in a context where people are facing an explosive increase in remote sensing image data. It is foreseeable that, in the future, the LFCF-ID will have wide engineering applications in urban study and planning.

Author Contributions: Conceptualization, B.C.; methodology, B.C. and Y.C.; software, Y.L.; validation, J.C.; formal analysis, B.C.; investigation, B.C., F.G. and Y.C.; resources, F.G.; data curation, F.G.; writing—original draft preparation, S.W.; writing—review and editing, C.Z.; supervision, C.Y.; visualization, F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (No.2017YFC08219) and the S&T Program of Hebei (21340302D).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We are grateful to the anonymous reviewers and academic editors whose constructive suggestions have improved the quality of this paper. Additionally, in the current severe epidemic situation of coronavirus disease 2019 (COVID-19), we would like to express our sincere gratitude to all frontline medical staff.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Li, D.; Tong, Q.; Li, R.; Gong, J.; Zhang, L. Some frontier scientific issues of high-resolution earth observation. *Sci. China Earth Sci.* 2012, 42, 805–813. [CrossRef]
- 2. Zhang, B. Current Status and Future Prospects of Remote Sensing. Bull. Chin. Acad. Sci. 2017, 32, 774–784.

- 3. Li, J.; Hu, Q.; Ai, M. Optimal Illumination and Color Consistency for Optical RemoteSensing Image Mosaicking. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1943–1947. [CrossRef]
- Sedaghat, A.; Ebadi, H. Remote Sensing Image Matching Based on Adaptive Binning SIFT Descriptor. *IEEE Trans. Geosci. Remote Sens.* 2015, 53, 5283–5293. [CrossRef]
- Xianyu, Z.; Minghao, X.; Xiangzhi, H.; Wenqian, Z.; Dongdong, S. A Full Coverage Retrieval Mode and Method for Remote Sensing Tile Data. J. Henan Univ. (Nat. Sci.) 2018, 48, 299–308.
- 6. Han, L.; Pong, G.; Jie, W.; Nicholas, C.; Yuqi, B.; Shunlin, L. Annual dynamics of global land cover and its long-term changes from 1982 to 2015. *Earth Syst. Sci. Data* **2020**, *2*, 1217–1243. [CrossRef]
- Zhang, X.; Liu, L.; Chen, X.; Gao, Y.; Xie, S.; Mi, J. GLC_FCS30: Global land-cover product with fine classification system at 30 m using time-series Landsat imagery. *Earth Syst. Sci. Data* 2021, 13, 2753–2776. [CrossRef]
- 8. Zhang, X.; Liu, L.; Chen, X.; Gao, Y.; Jiang, M. Automatically Monitoring Impervious Surfaces Using Spectral Generalization and Time Series Landsat Imagery from 1985 to 2020 in the Yangtze River Delta. *J. Remote Sens.* **2021**, 2021, 873816. [CrossRef]
- 9. Wang, D.L.; Hu, F. Monitoring of illegal buildings in Beijing using high-resolution satellite imagery from China. *Chin. Sci. Bull.* **2009**, *54*, 305–311. [CrossRef]
- Moghadam, N.K.; Delavar, M.R.; Hanachee, P. Automatic urban illegal building detection using multi-temporal satellite images and geospatial information systems. *ISPRS—Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 2015, XL-1-W5, 387–393. [CrossRef]
- 11. Hu, F.; Xia, G.S.; Zhang, L. Deep sparse representations for land-use scene classification in remote sensing images. In Proceedings of the 2016 IEEE 13th International Conference on Signal Processing (ICSP), Chengdu, China, 6–10 November 2016.
- 12. Li, Y.; Zhang, Y.; Huang, X.; Zhu, H.; Ma, J. Large-Scale Remote Sensing Image Retrieval by Deep Hashing Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 950–965. [CrossRef]
- 13. Li, J.; Shen, B.; Jiang, R.; Chen, T. Quadtree Spatial Index Algorithm for Image Pyramid. Comput. Eng. 2011, 37, 11–13.
- 14. Cheng, G.; Han, J.W.; Lu, X.Q. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE* 2017, 105, 1865–1883. [CrossRef]
- 15. Egenhofer, M.J. Spatial-Query-by-Sketch. In Proceedings of the IEEE Symposium on Visual Languages, Washington, DC, USA, 3 October 1996.
- 16. Lee, Y.C.; Chin, F.L. An iconic query language for topological relationships in GIS. Int. J. Geogr. Inf. Syst. 1995, 9, 25–46. [CrossRef]
- 17. Shekhar, S.; Chawla, S.; Ravada, S.; Fetterer, A.; Liu, X.; Lu, C. Spatial databases-accomplishments and research needs. Knowledge and Data Engineering. *IEEE Trans.* **1999**, *11*, 45–55.
- 18. Gaede, V.; Günther, O. Multidimensional access methods. ACM Comput. Surv. 1998, 30, 170–231. [CrossRef]
- Aptoula, E. Remote Sensing Image Retrieval with Global Morphological Texture Descriptors. *IEEE Trans. Geosci. Remote Sens.* 2014, 52, 3023–3034. [CrossRef]
- Li, F.; You, S.; Wei, H.; Wei, E.; Chen, L. Filtering model of optimal dataset for remote sensing image area coverage. *Radio Eng.* 2017, 47, 45–48.
- 21. Maulik, U.; Bandyopadhyay, S. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1075–1081. [CrossRef]
- Stavrakoudis, D.; Theocharis, J.; Zalidis, G. A Boosted Genetic Fuzzy Classifier for land cover classification of remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* 2011, 66, 529–544. [CrossRef]
- 23. Tseng, M.-H.; Chen, S.-J.; Hwang, G.-H.; Shen, M.-Y. A genetic algorithm rule-based approach for land-cover classification. *ISPRS J. Photogramm. Remote Sens.* 2008, 63, 202–212. [CrossRef]
- 24. Nikfar, M.; Zoej, M.J.V.; Mohammadzadeh, J.; Mokhtarzade, M.; Navabi, A. Optimization of multiresolution segmentation by using a genetic algorithm. J. Appl. Remote Sens. 2012, 6, 063592. [CrossRef]
- 25. Krawiec, K.; Bhanu, B. Visual learning by coevolutionary feature synthesis. *IEEE Trans. Syst. Man Cybern. Part B (Cybernetics)* 2005, *35*, 409–425. [CrossRef]
- 26. Pedergnana, M.; Marpu, P.R.; Mura, M.D.; Benediktsson, J.A.; Bruzzone, L. A Novel Technique for Optimal Feature Selection in Attribute Profiles Based on Genetic Algorithms. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 3514–3528. [CrossRef]
- 27. Puig, D.; Garcia, M.A. Automatic texture feature selection for image pixel classification. *Pattern Recognit.* **2006**, *39*, 1996–2009. [CrossRef]
- 28. Ines, A.V.; Honda, K. On quantifying agricultural and water management practices from low spatial resolution RS data using genetic algorithms: A numerical study for mixed-pixel environment. *Adv. Water Resour.* **2005**, *28*, 856–870. [CrossRef]
- 29. Zhan, H.; Lee, Z.; Shi, P.; Chen, C.; Carder, K. Retrieval of water optical properties for optically deep waters using genetic algorithms. *IEEE Trans. Geosci. Remote Sens.* 2003, 41, 1123–1128. [CrossRef]